Understanding Testing Strategies

Multiple dimensions of quality

Testing is measurement

The better the question, the better the answer

The Performance Measure

Measuring performance is load-testing

How does it perform under load?

What is the maximum load, given a performance measure?

Load Testing Production

Load testing is dynamic testing, so it requires deployment

"If you're not load testing production, you're not load testing production"

Otherwise, you're only gaining knowledge about your test environment

How Can I Load Test Production?

Deep-night testing

It's 10AM somewhere

An environment which will *become* Production

Apache JMeter

Platform doesn't matter here

Your code isn't tied up in this

Working with JMeter

JMX file

Created in a GUI or programmatically

Robustness

The degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions

Fuzz Testing

OneFuzz

A sibling to OWASP ZAP

The Availability Measure

Availability is a measure of your infrastructure

Though your infrastructure should be a consequence of your code

Postman

Integration testing – how systems interact

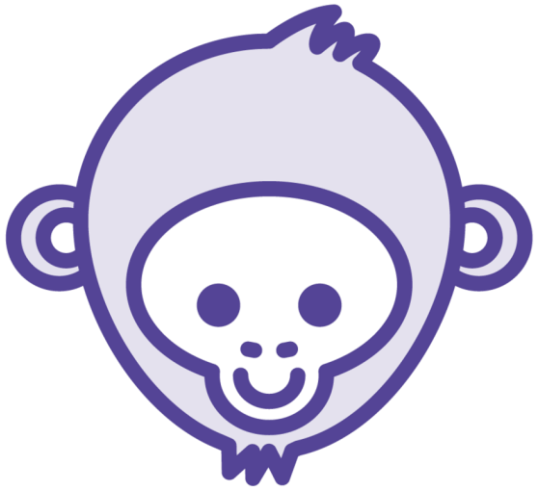Postman is pleasingly close to the metal
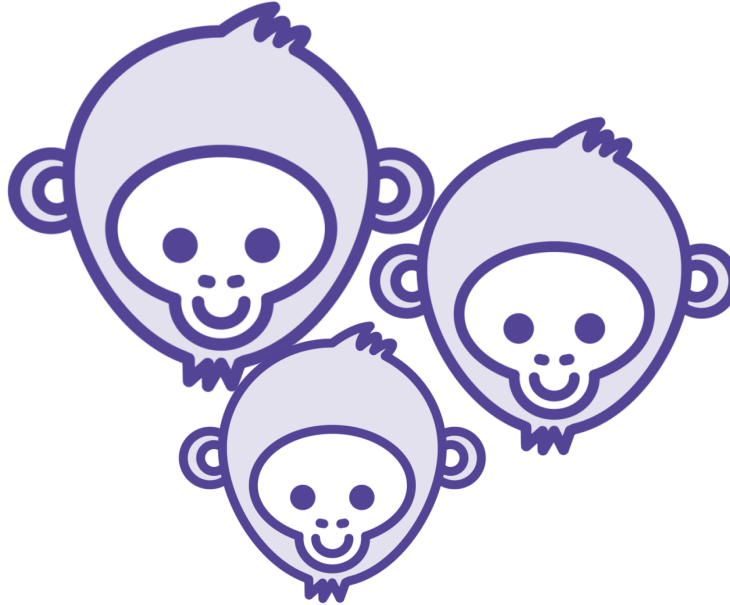
Netflix and Chaos Testing

- **Creating unexpected circumstances…and seeing what happens**
- **Human beings imagine failure poorly**
- **"An ounce of test is worth a pound of guess"**
- **Different aspects of chaos**

The Chaos Monkey

The engine of chaos

The Simian Army

Not a tool, but a practice

Connecting Your Processes Together

Disparate toolsets

Especially while you're migrating

Service Connections and Service Hooks

Two broad ways of connecting

Service hooks

Service connections

The event originator manages the event hook.

Integrating Github with Azure Pipelines

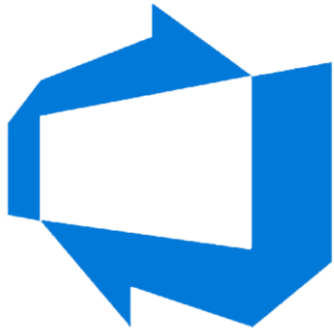There are exceptions, especially when things get more complex

Code is in Github, build is in Azure DevOps

Everybody knows Github, so we're not changing anytime soon

Further Integration with Azure

The builds are in DevOps

Maybe the code is in Azure, and the build is in Jenkins

In that case, DevOps is just another remote

Linking Git Work Items

Linking work items is powerful

Once you're linked, you just reference the work item id in the commit message

The fact that it's bound for GitHub doesn't matter

GitHub Actions

- YAML-defined
- Different tasks, including build and deployment

Course Summary

Build Tools
- SonarQube
- For static analysis
  - WhiteSource Bolt
  - BlackDuck
- For software composition analysis
  - OWASP Zap
  - OneFuzz
    - For security and Fuzz testing

Unit Test Coverage
- How to enforce that in a build

Integration with Github
- In both directions